Towards Temporal Mobility Markov Chains

Sébastien Gambs, Marc-Olivier Killijian, Miguel Nuñez del Prado Cortez

Towards Temporal Mobility Markov Chains

Sébastien Gambs
Université de Rennes 1 - INRIA / IRISA ;
Campus Universitaire de Beaulieu
35042 Rennes, France

Marc-Olivier Killijian
Miguel Núñez del Prado Cortez
CNRS ; LAAS, 7 avenue du Colonel Roche
F-31077 Toulouse, France
Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS
marco.killijian@laas.fr

I. Mobility Markov Chains

In [1], we defined a type of mobility model that we coined as *mobility Markov chain (MMC)*, which can represent in a compact yet precise way the mobility behaviour of an individual. In short, a MMC is a probabilistic automaton in which states represent points of interest (POIs) of an individual and transitions between states corresponds to a movement from one POI to another one. The automaton is probabilistic in the sense that a transition between POIs is non deterministic but rather that there is a probability distribution over the transitions that corresponds to the probability of moving from one POI to another (edges are directed). Note that Markov models are a popular technique that have been used in the past for the study of motion (for instance see [2] for a recent work using hidden Markov networks to extract POIs from geolocated data).

More formally, a MMC is a transition system composed of:

- *A set of states* $P = \{p_1, \ldots, p_n\}$, in which each state p_i corresponds to a POI (or a set of POIs). These POIs may have been learned for instance by running a clustering algorithm on the trail of mobility traces from an individual or simply by collecting the locations that he has posted on a geolocated social network such as Foursquare or Gowalla. Each state (*i.e.* POI) is therefore associated with a physical location.Moreover, it is often possible to attach a *semantic label* to the states of the mobility Markov chain, such as for instance "home" instead of simply p_1 or "work" instead of p_2.
- *A set of transitions*, $T = \{t_{1,1}, \ldots, t_{n,n}\}$, where each transition $t_{i,j}$ represents a movement from the state p_i to the state p_j. Each transition $t_{i,j}$ has a probability assigned to it that corresponds to the probability of moving from state p_i to state p_j. Sometimes an individual can move from one POI, go somewhere else (but not to one of his usual POIs) and come back later to the same POI. For example, an individual might leave his house to go wash his car in a facility near his home and come back 30 minutes later. This type of behaviour is materialized in the mobility Markov chain by a transition from one state to itself.

A MMC can be represented either as a *transition matrix* or as a *directed graph* in which nodes correspond to states and there is a directed weighted edge between two nodes if and only if the transition probability between these two nodes is non-null. The sum of all the edges' weights leaving from a state is equal to 1.

For instance, consider for illustration purpose, an individual, that we refer thereafter as "Alice", who has a set of 4 important POIs that she visits often plus some other POIs that are less important to her mobility. In order to represent her mobility behaviour, we could define a MMC composed of 5 states, one for each important POI plus a last one that will contain all the infrequent POIs. Suppose now that we have been able to collect her trail of mobility traces (*e.g.*, by tracking her mobile phone [3]), then possibly we could have learnt the following MMC (Figure 1). For more details, about MMC we refer the reader to [1].

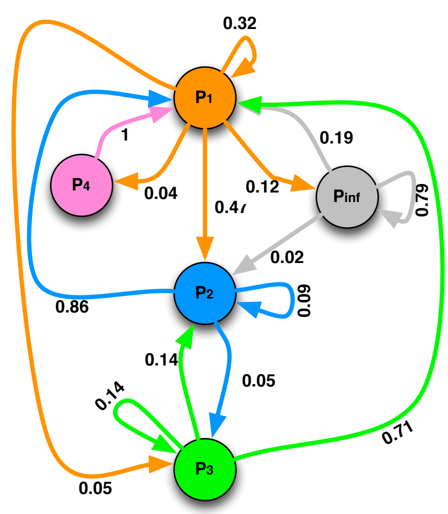

Fig. 1. Alice's Mobility Markov Chain.

II. Introducing Time into Mobility Markov Chains

In this section, we describe an algorithm for learning Temporal Mobility Markov Chains (TMMC). This algorithm is an extension of the one described in [1] to which we add the temporal aspect. Basically, the gist of this algorithm is to decompose time into n different discrete timeslices. For example, with $n = 4$, we can observe mobility in one of the following timeslices: *morning*, *afternoon*, *evening* and *night*. Each POI from the original MMC will thus be represented by n different states in the TMMC, one for each timeslice. If we

Algorithm 1 Temporal mobility Markov chain algorithm

Require: Trail of (mobility) traces M, merging distance d, speed threshold ϵ, time interval threshold $mintime$, time slices definition $timeslices$

1: Run a clustering algorithm on M to learn the most significant clusters
2: Merge all the clusters that are within d distance of each other
3: Let $listPOIs$ be the list of all remaining clusters
4: **for** each cluster C in $listPOIs$ **do**
5: Compute the $time\_interval$ and the $density$ of C
6: **end for**
7: **for** each cluster C in $listPOIs$ **do**
8: **if** $C.time\_interval > mintime$ **then**
9: Add C to $freqPOIs$ (the list of frequent POIs)
10: **else**
11: Add C to $infreqPOIs$ (the list of infrequent POIs)
12: **end if**
13: **end for**
14: Sort the clusters in $freqPOIs$ by decreasing order according to their densities
15: **for** each cluster C_i in $freqPOIs$ (for $1 \leq i \leq n-1$) **do**
16: **for** each time slice t in $timeslices$ **do**
17: Create a state $p_{i/t}$ in the mobility Markov chain
18: **end for**
19: **end for**
20: Create a state $p_{infrequent}$ representing all the clusters within $infreqPOIs$
21: Let M' be the trail of traces obtained from M by removing all the traces whose speed is above ϵ
22: **for** each mobility trace in M' **do**
23: **if** the distance between the trace and the state p_i is less than d and the state p_i is the closest state **then**
24: determine t the appropriate time slice for the trace
25: label the trace with "$p_{i/t}$"
26: **else**
27: label the state with the value "unknown"
28: **end if**
29: **end for**
30: Squash all the successive mobility traces sharing the same (time and space) label into a single occurrence
31: Compute all the transition probabilities between each pair of states of the Markov chain
32: **return** the mobility Markov chain computed

take Alice's home for example, which is state p_1 on Figure 1., it will be represented by the states $P_{1morning}$, $P_{1afternoon}$, $P_{1evening}$ and P_{1night}. Remark that the scale of the timeslicing can be tuned to match the required level of detail. For example, the timeslices can also be $winter, spring, summer, autumn$ or even months or years, ...

In a nutshell, Algorithm 1 starts (line 1) by applying a clustering algorithm on a trail of traces of an individual in order to identify clusters of locations that are significant. Then, in order to reduce the number of resulting clusters, the algorithm merges clusters whose medoids are within a predefined distance d of each other (line 2). Each resulting cluster is considered as a POI, and the medoid of the cluster is considered to be the physical location of the POI. For each cluster (lines 4 to 6), we compute the number of mobility traces inside the cluster (which we call the *density* of the cluster) and the *time interval* (measured in days) between the earliest and the latest mobility traces of the cluster (line 5). The POIs (*i.e.* clusters) are then split (lines 7 to 13) into two categories; the *frequent POIs* that correspond to POIs whose time interval is above or equal to a certain threshold $mintime$ and the *infrequent POIs* whose time interval is below this threshold $mintime$. In the set of frequent POIs (line 14), we sort the POIs by decreasing order according to their densities. Therefore, the first POI will be the denser and the last POI the less dense.

Now, we can start to build the temporal mobility Markov chain by creating a state for each tuple $< POI, timeslice >$ within the set of frequent POIs (lines 15 to 19) and also a last state representing the set of infrequent POIs (line 20). Each state is then assigned a weight that is set to its density. Afterwards (line 21), we come back to the trail of traces that have been used to learn the POIs and we remove all the moving points (whose speed is above ϵ, for ϵ a small predefined value). Then, we traverse the trail of traces in a chronological order (lines 22 to 29) labeling each of the mobility traces either with the tag of closest POI and the appropriate timeslice (line 25) or with the tag "unknown" if the mobility trace is not within d-distance of one of the frequent or infrequent POIs (line 27). From this labeling, we can extract sequences spatio-temporal chunks that have been visited by the individual in which all the successive mobility traces sharing the same label are merged into a single occurrence (line 30). For example, a typical day could be summarized by the following sequence "$p_{home/morning} \Rightarrow p_{work/morning} \Rightarrow p_{work/afternoon} \Rightarrow p_{sport/afternoon} \Rightarrow p_{sport/evening} \Rightarrow$ "$unknown''$ $\Rightarrow p_{home/evening}$". From the set of sequences extracted, we compute the transition probabilities between the different states of the MMC by counting the number of transitions between each pair of states and then normalizing these probabilities (line 31). If we observe a subsequence in the form of "$p_{i/t} \Rightarrow$ "unknown" $\Rightarrow p_{i/t}$" then we increment the count from the state $p_{i/t}$ to itself (which translates in the graph representation by a self-arrow).

References

[1] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Show me how you move and i will tell you who you are," *Transactions on Data Privacy*, vol. 4, no. 2, pp. 103–126, 2011.

[2] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories," in *14th International Conference on Extending Database Technology (EDBT)*, 2011.

[3] M.-O. Killijian, M. Roy, and G. Trédan, "Beyond san francisco cabs : Building a *-lity mining dataset," in *Workshop on the Analysis of Mobile Phone Networks (NetMob 2010)*, 2010.